

Data Structures Day 3

—

Soumajit Pramanik

Applications of Stacks

Mathematical Calculations

What does $3 + 2 * 4$ equal?

$2 * 4 + 3$? $(3 + 2) * 4$?

The precedence of operators affects the order of operations.

A mathematical expression cannot simply be evaluated left to right.

A challenge when evaluating a program.

Lexical analysis is the process of interpreting a program.

What about $1 - 2 - 4 ^ 5 * 3 * 6 / 7 ^ 2 ^ 3$

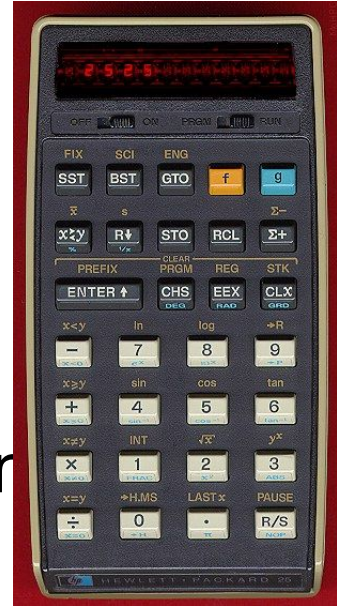
Infix and Postfix Expressions

The way we are use to writing expressions is known as infix notation

Postfix expression does not require any precedence rules

3 2 * 1 + is postfix of 3 * 2 + 1

evaluate the following postfix expression
and write out a corresponding infix
expression:


$$2^3 2^4 * + *$$

1 2 3 4 ^ * +

$$12 - 32^3 * 6 / +$$
 $25^1 -$

Infix to postfix conversion

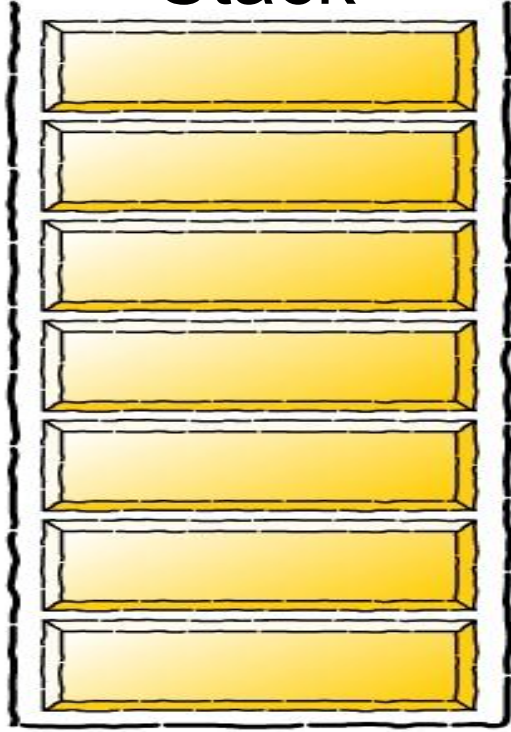
Scan the Infix expression left to right

- If the character x is an operand
 - Output the character into the Postfix Expression
- If the *character* x is a left or right parenthesis
 - If the character is “(“
 - Push it into the stack
 - if the *character* is “)”
 - Repeatedly pop and output all the operators/characters until “(“ is popped from the stack.
- If the character x is a regular operator
 - **Step 1:** Check the character y currently at the top of the stack.
 - **Step 2:** If Stack is empty or $y = '('$ or y is an operator of lower precedence than x , then push x into stack.
 - **Step 3:** If y is an operator of higher or equal precedence than x , then pop and output y and push x into the stack.

When all characters in infix expression are processed repeatedly pop the character(s) from the stack and output them until the stack is empty.

Infix to postfix conversion

Stack



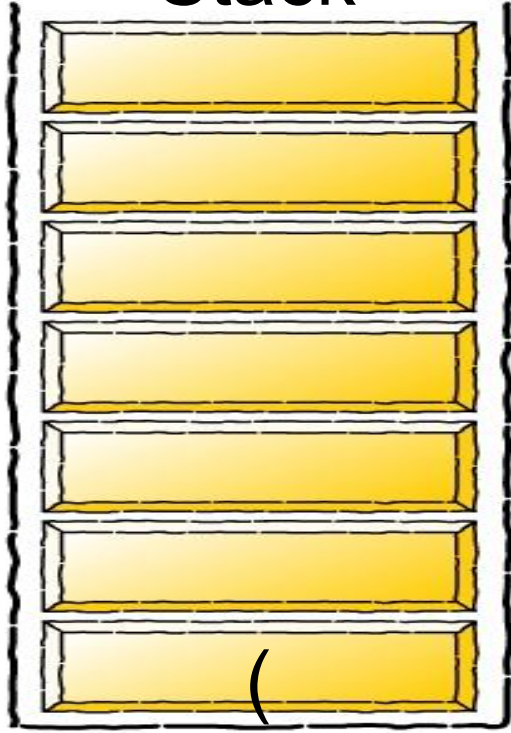
Infix Expression

$(a + b - c) * d - (e + f)$

Postfix Expression

Infix to postfix conversion

Stack



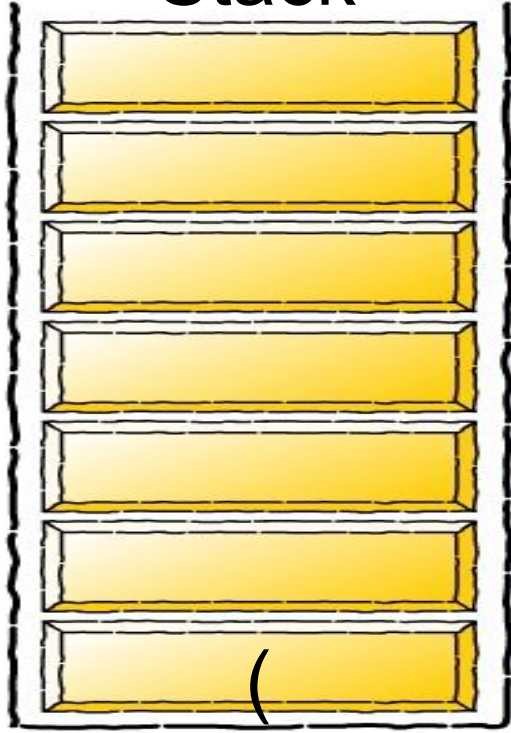
Infix Expression

$a + b - c) * d - (e + f)$

Postfix Expression

Infix to postfix conversion

Stack



Infix Expression

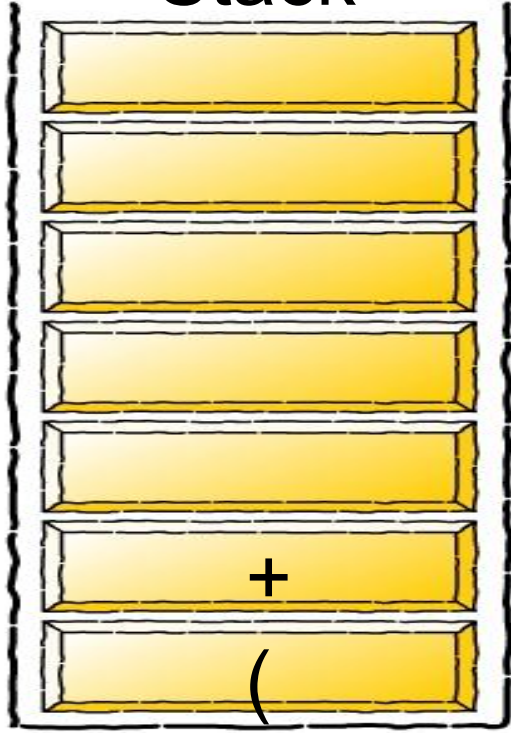
$+ b - c) * d - (e + f)$

Postfix Expression

a

Infix to postfix conversion

Stack



Infix Expression

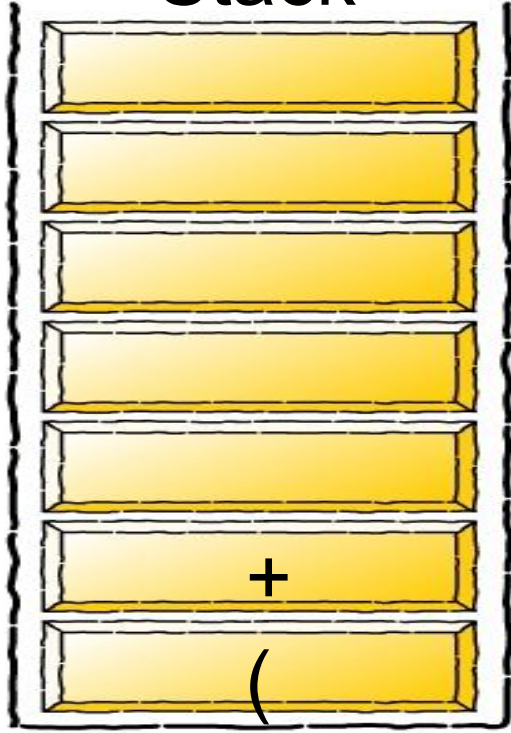
$b - c) * d - (e + f)$

Postfix Expression

a

Infix to postfix conversion

Stack



Infix Expression

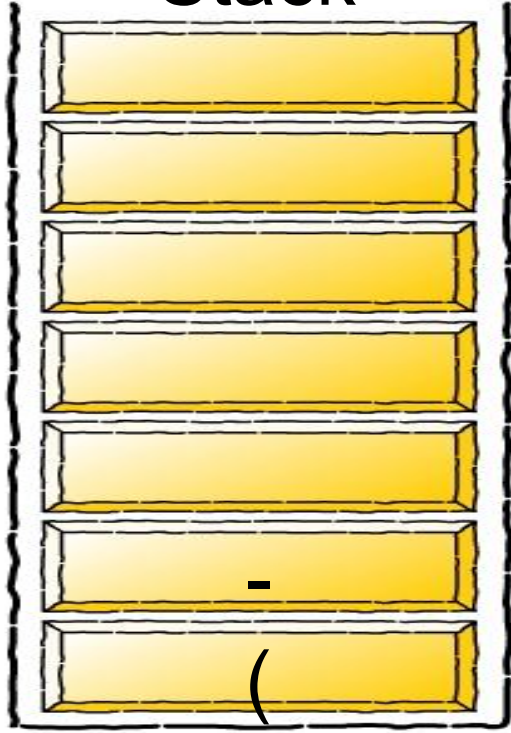
$- c) * d - (e + f)$

Postfix Expression

$a b$

Infix to postfix conversion

Stack



Infix Expression

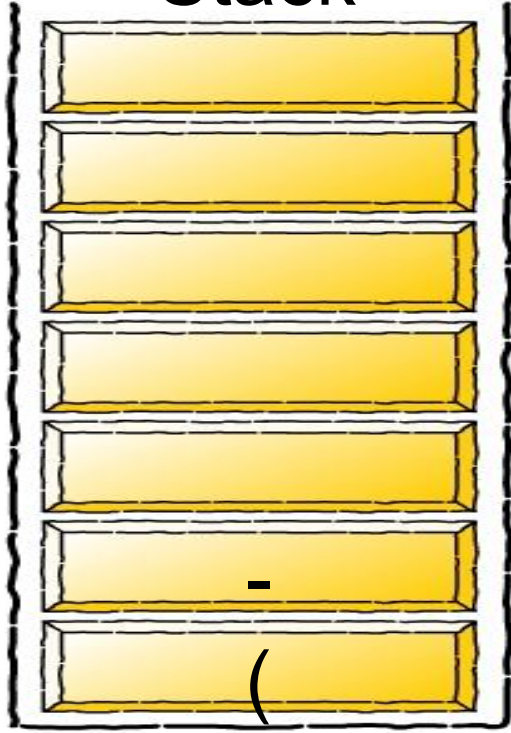
$c) * d - (e + f)$

Postfix Expression

$a b +$

Infix to postfix conversion

Stack



Infix Expression

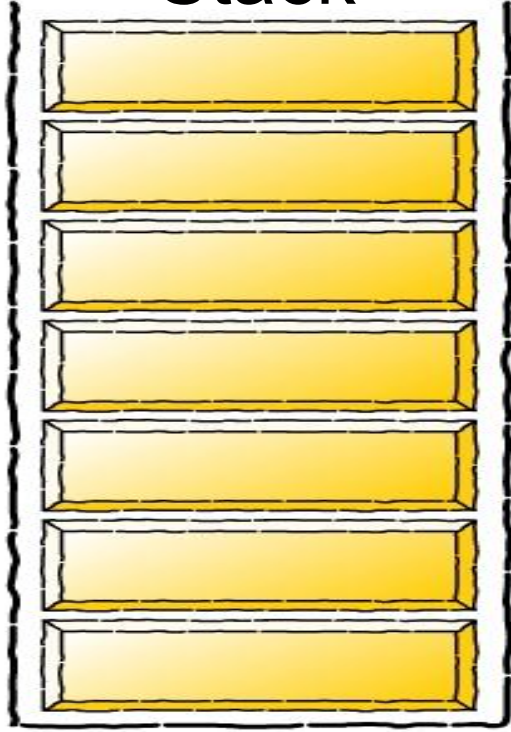
$) * d - (e + f)$

Postfix Expression

$a b + c$

Infix to postfix conversion

Stack



Infix Expression

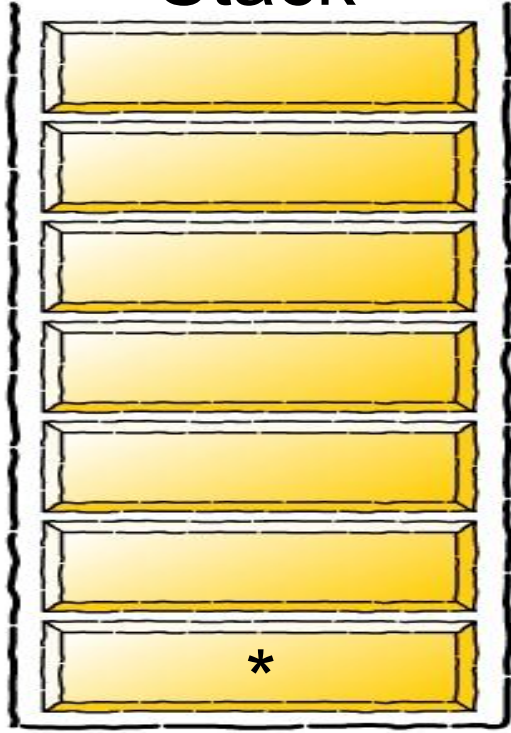
$* d - (e + f)$

Postfix Expression

$a b + c -$

Infix to postfix conversion

Stack



Infix Expression

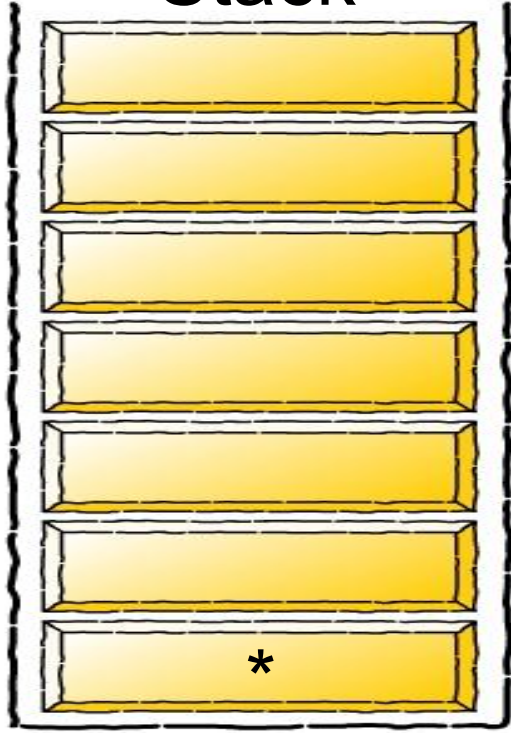
$d - (e + f)$

Postfix Expression

$a b + c -$

Infix to postfix conversion

Stack



Infix Expression

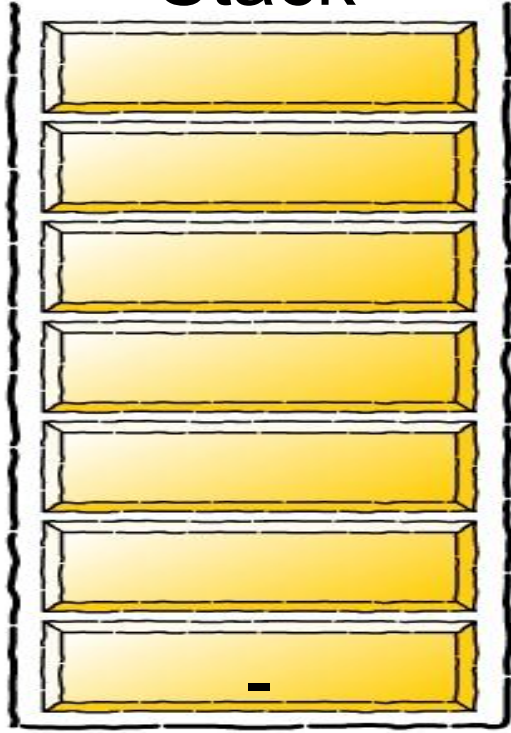
$- (e + f)$

Postfix Expression

$a b + c - d$

Infix to postfix conversion

Stack



Infix Expression

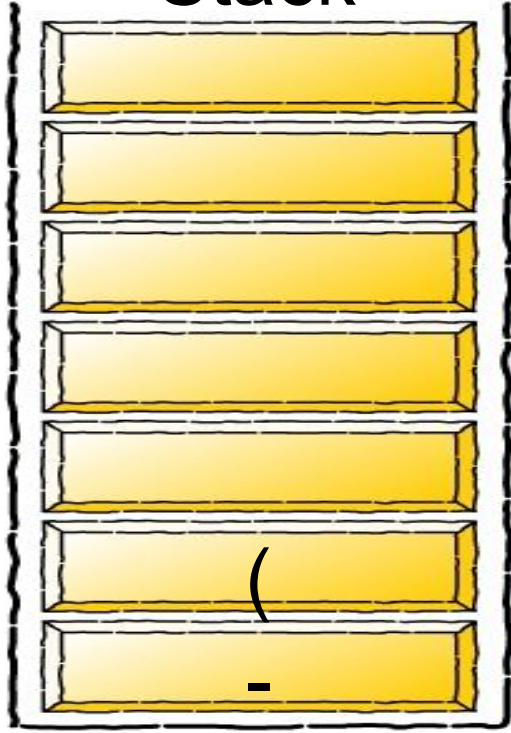
(e + f)

Postfix Expression

a b + c - d *

Infix to postfix conversion

Stack



Infix Expression

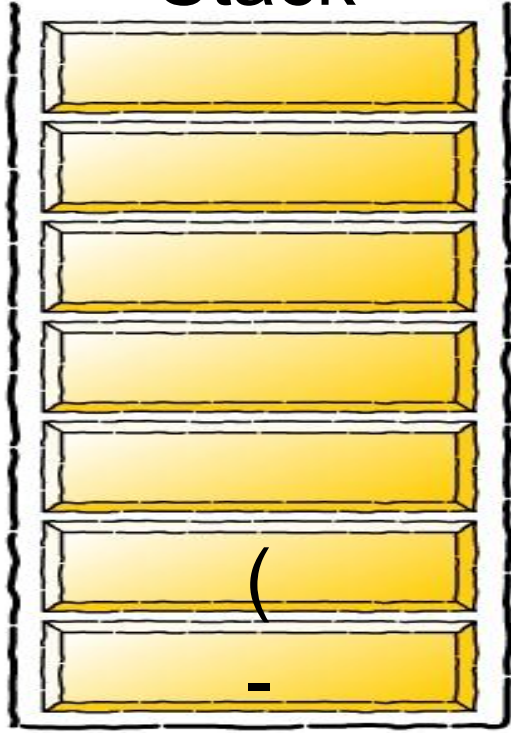
$e + f)$

Postfix Expression

$a b + c - d *$

Infix to postfix conversion

Stack



Infix Expression

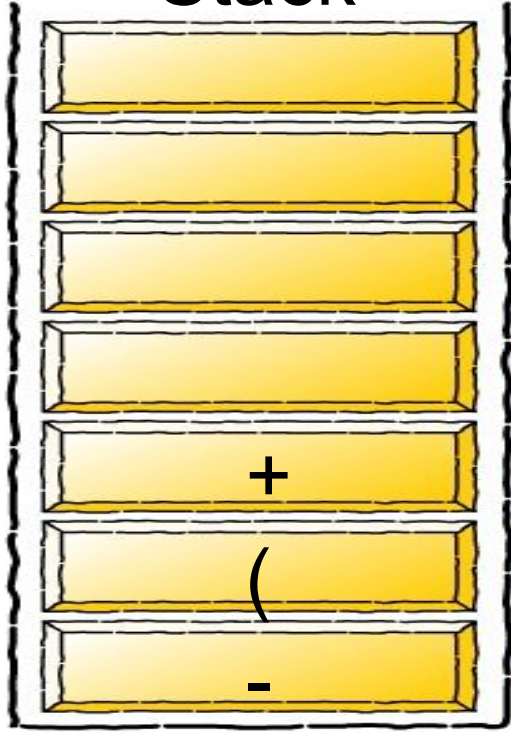
+ f)

Postfix Expression

a b + c - d * e

Infix to postfix conversion

Stack



Infix Expression

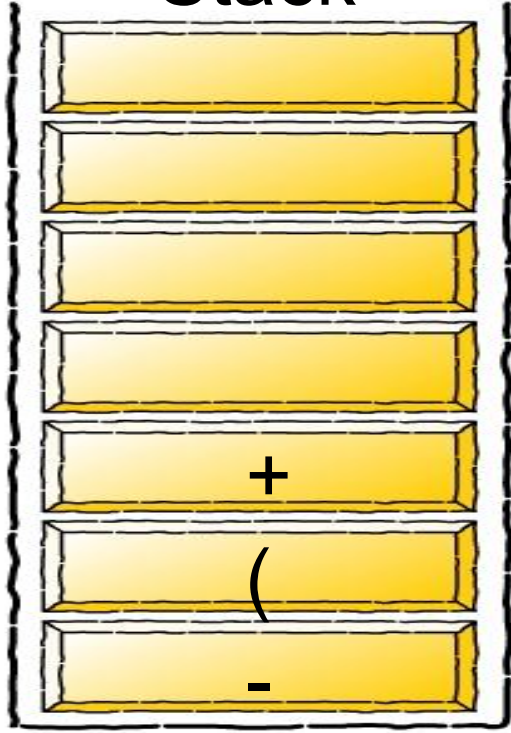
f)

Postfix Expression

a b + c - d * e

Infix to postfix conversion

Stack



Infix Expression

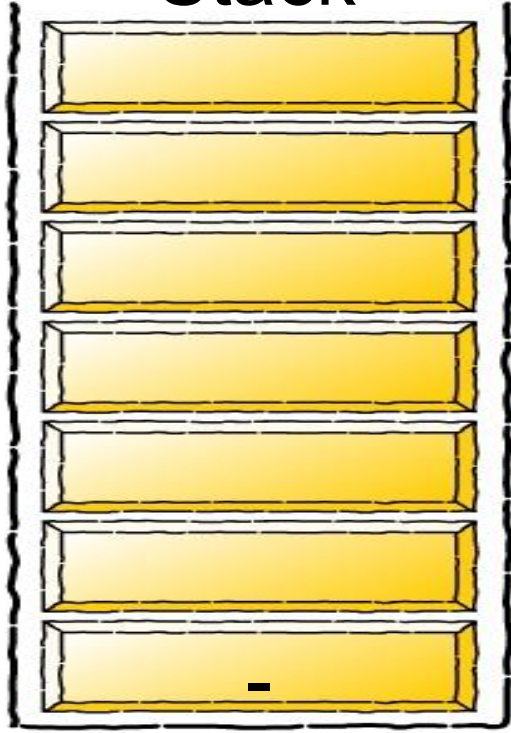
)

Postfix Expression

a b + c - d * e f

Infix to postfix conversion

Stack



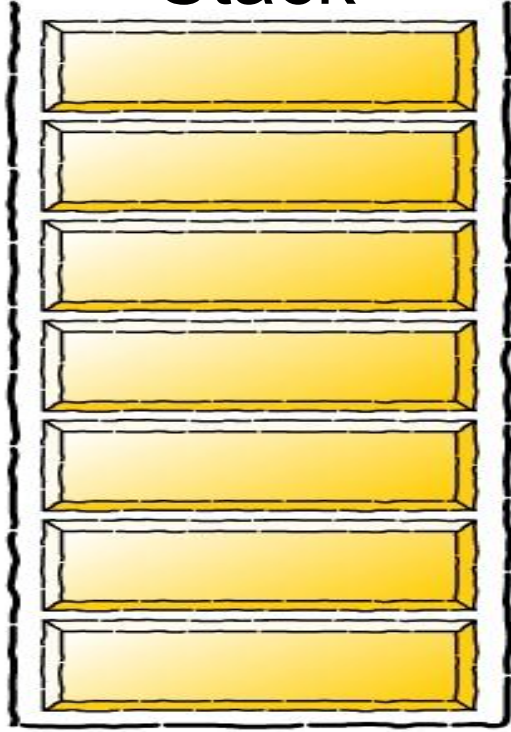
Infix Expression

Postfix Expression

a b + c - d * e f +

Infix to postfix conversion

Stack



Infix Expression

Postfix Expression

a b + c - d * e f + -